Probabilistic Face Embeddings

Yichun Shi and Anil K. Jain
Michigan State University, East Lansing, MI
shiyichu@msu.edu, jain@cse.msu.edu

Abstract

Embedding methods have achieved success in face recognition by comparing facial features in a latent semantic space. However, in a fully unconstrained face setting, the facial features learned by the embedding model could be ambiguous or may not even be present in the input face, leading to noisy representations. We propose Probabilistic Face Embeddings (PFEs), which represent each face image as a Gaussian distribution in the latent space. The mean of the distribution estimates the most likely feature values while the variance shows the uncertainty in the feature values. Probabilistic solutions can then be naturally derived for matching and fusing PFEs using the uncertainty information. Empirical evaluation on different baseline models, training datasets and benchmarks show that the proposed method can improve the face recognition performance of deterministic embeddings by converting them into PFEs. The uncertainties estimated by PFEs also serve as good indicators of the potential matching accuracy, which are important for a risk-controlled recognition system.

1. Introduction

When humans are asked to describe a face image, they not only give the description of the facial attributes, but also the confidence associated with them. For example, if the eyes are blurred in the image, a person will keep the eye size as an uncertain information and focus on other features. Furthermore, if the image is completely corrupted and no attributes can be discerned, the subject may respond that he/her cannot identify this face. This kind of uncertainty (or confidence) estimation is common and important in human decision making.

On the other hand, the representations used in state-of-the-art face recognition systems are generally confidence-agnostic. These methods depend on an embedding model (e.g. Deep Neural Networks) to give a deterministic point representation for each face image in the latent feature space [27, 35, 20, 34, 4]. A point in the latent space represents the model’s estimation of the facial features in the given image. If the error in the estimation is somehow bounded, the distance between two points can effectively measure the semantic similarity between the corresponding face images. But given a low-quality input, where the expected facial features are ambiguous or absent in the image, a large shift in the embedded points is inevitable, leading to false recognition (Figure 1a).

Figure 1: Difference between deterministic face embeddings and probabilistic face embeddings (PFEs). Deterministic embeddings represent every face as a point in the latent space without regards to its feature ambiguity. Probabilistic face embedding (PFE) gives a distributional estimation of features in the latent space instead. Best viewed in color.

Given that face recognition systems have already achieved high recognition accuracies on relatively constrained face recognition benchmarks, e.g. LFW [10] and YTF [37], where most facial attributes can be clearly observed, recent face recognition challenges have moved on to more unconstrained scenarios, including surveillance videos [18, 23, 12] (See Figure 2). In these tasks, any type and degree of variation could exist in the face image, where most of the desired facial features learned by the representation model could be absent. Given this lack of information, it is unlikely to find a feature set that could always match these faces accurately. Hence state-of-the-art face recognition systems which obtained over 99% accuracy on LFW have suffered from a large performance drop on IARPA Janus benchmarks [18, 23, 12].

To address the above problems, we propose Probabilistic Face Embeddings (PFEs), which give a distributional estimation instead of a point estimation in the latent space for each input face image (Figure 1b). The mean of the distribution can be interpreted as the most likely latent feature values while the span of the distribution represents the un-
they are not suitable for our task since our target space is not well-defined by given labels\(^2\). Variational Autoencoders [17] can also be regarded as a method for estimating data uncertainty, but it mainly serves a generation purpose. Specific to face recognition, some studies [6, 16, 45] have leveraged the model uncertainty for analysis and learning of face representations, but to our knowledge, ours is the first work that utilizes data uncertainty\(^1\) for recognition tasks.

**Probabilistic Face Representation** Modeling faces as probabilistic distributions is not a new idea. In the field of face template/video matching, there exists abundant literature on modeling the faces as probabilistic distributions [29, 1], subspace [3] or manifolds [1, 11] in the feature space. However, the input for such methods is a set of face images rather than a single face image, and they use a between-distribution similarity or distance measure, e.g. KL-divergence, for comparison, which does not penalize the uncertainty. Meanwhile, some studies [19, 9] have attempted to build a fuzzy model of a given face using the features of face parts. In comparison, the proposed PFE represents each single face image as a distribution in the latent space encoded by DNNs and we use an uncertainty-aware log likelihood score to compare the distributions.

**Quality-aware Pooling** In contrast to the methods above, recent work on face template/video matching aims to leverage the saliency of deep CNN embeddings by aggregating the deep features of all faces into a single compact vector [41, 21, 39, 7]. In these methods, a separate module learns to predict the quality of each face in the image set, which is then normalized for a weighted pooling of feature vectors. We show that a solution can be naturally derived under our framework, which not only gives a probabilistic explanation for quality-aware pooling methods, but also leads to a more general solution where an image set can also be modeled as a PFE representation.

3. **Limitations of Deterministic Embeddings**

In this section, we explain the problems of deterministic face embeddings from both theoretical and empirical views. Let \(X\) denote the image space and \(Z\) denote the latent feature space of \(D\) dimensions. An ideal latent space \(Z\) should only encode identity-salient features and be disentangled from identity-irrelevant features. As such, each identity should have a unique intrinsic code \(z \in Z\) that best represents this person and each face image \(x \in X\) is an observation sampled from \(p(x|z)\). The process of training face embeddings can be viewed as a joint process of searching for such a latent space \(Z\) and learning the inverse mapping

\(^1\)https://github.com/seasonSH/Probabilistic-Face-Embeddings

\(^2\)Although we are given the identity labels, they cannot directly serve as target vectors in the latent feature space.

\(^3\)Some in the literature have also used the terminology “data uncertainty” for a different purpose [40].
Figure 3: Illustration of feature ambiguity dilemma. The plots show the cosine similarity on LFW dataset with different degrees of degradation. Blue lines show the similarity between original images and their respective degraded versions. Red lines show the similarity between impostor pairs of degraded images. The shading indicates the standard deviation. With larger degrees of degradation, the model becomes more confident (very high/low scores) in a wrong way.

$p(z|x)$. For deterministic embeddings, the inverse mapping is a Dirac delta function $p(z|x) = \delta(z - f(x))$, where $f$ is the embedding function. Clearly, for any space $Z$, given the possibility of noises in $x$, it is unrealistic to recover the exact $z$ and the embedded point of a low-quality input would inevitably shift away from its intrinsic $z$ (no matter how much training data we have).

The question is whether this shift could be bounded such that we still have smaller intra-class distances compared to inter-class distances. However, this is unrealistic for fully unconstrained face recognition and we conduct an experiment to illustrate this. Let us start with a simple example: given a pair of identical images, a deterministic embedding will always map them to the same point and therefore the distance between them will always be 0, even if these images do not contain a face. This implies that “a pair of images being similar or even the same does not necessarily mean the probability of their belonging to the same person is high”.

To demonstrate this, we conduct an experiment by manually degrading the high-quality images and visualizing their similarity scores. We randomly select a high-quality image of each subject from the LFW dataset [10] and manually insert Gaussian blur, occlusion, and random Gaussian noise to the faces. In particular, we linearly increase the size of Gaussian kernel, occlusion ratio and the standard deviation of the noise to control the degradation degree. At each degradation level, we extract the feature vectors with a 64-layer CNN\(^4\), which is comparable to state-of-the-art face recognition systems. The features are normalized to a hyper-spherical embedding space. Then, two types of cosine similarities are reported: (1) similarity between pairs of original images and its respective degraded image, and (2) similarity between degraded images of different identities. As shown in Figure 3, for all the three types of degradation, the genuine similarity scores decrease to 0 while the impostor similarity scores converge to 1.0! These indicate two types of errors that can be expected in a fully unconstrained scenario even when the model is very confident (very high/low similarity scores):

(1) false accept of impostor low-quality pairs and
(2) false reject of genuine cross-quality pairs.

To confirm this, we test the model on the IJB-A dataset by finding impostor/genuine image pairs with the highest/lowest scores, respectively. The situation is exactly as we hypothesized (See Figure 4). We call this Feature Ambiguity Dilemma which is observed when the deterministic embeddings are forced to estimate the features of ambiguous faces. The experiment also implies that there exist a dark space where the ambiguous inputs are mapped to and the distance metric is distorted.

4. Probabilistic Face Embeddings

To address the aforementioned problem caused by data uncertainty, we propose to encode the uncertainty into the face representation and take it into account during matching. Specifically, instead of building a model that gives a point estimation in the latent space, we estimate a distribution $p(z|x)$ in the latent space to represent the potential
In particular, we use a multivariate Gaussian distribution:

\[ p(z|x_i) = \mathcal{N}(z; \mu_i, \sigma_i^2 \mathbf{I}) \]  

(1)

where \( \mu_i \) and \( \sigma_i \) are both a \( D \)-dimensional vector predicted by the network from the \( i \)th input image \( x_i \). Here we only consider a diagonal covariance matrix to reduce the complexity of the face representation. This representation should have the following properties:

1. The center \( \mu \) should encode the most likely facial features of the input image.
2. The uncertainty \( \sigma \) should encode the model’s confidence along each feature dimension.

In addition, we wish to use a single network to predict the distribution. Considering that new approaches for training face embeddings are still being developed, we aim to develop a method that could convert existing deterministic face embedding networks to PFEs in an easy manner. In the followings, we first show how to compare and fuse the PFE representations to demonstrate their strength and then propose our method for learning PFEs.

**Note 1.** Because of the space limit, we provide the proofs of all the propositions below in the Supplementary Material.

### 4.1. Matching with PFEs

Given the PFE representations of a pair of images \( (x_i, x_j) \), we can directly measure the “likelihood” of them belonging to the same person (sharing the same latent code):

\[ p(z_i = z_j), \quad \text{where} \quad z_i \sim p(z|x_i) \text{ and } z_j \sim p(z|x_j). \]

Specifically,

\[ p(z_i = z_j) = \int p(z_i|x_i)p(z_j|x_j)\delta(z_i - z_j)dz_idz_j. \]  

(2)

In practice, we would like to use the log likelihood instead, whose solution is given by:

\[ s(x_i, x_j) = \log p(z_i = z_j) \]

\[ = -\frac{1}{2} \sum_{l=1}^{D} \left( \frac{(\mu_i^{(l)} - \mu_j^{(l)})^2}{\sigma_i^{2(l)} + \sigma_j^{2(l)}} + \log(\sigma_i^{2(l)} + \sigma_j^{2(l)}) \right) \]

\[ - \text{const}, \]  

(3)

where \( \text{const} = \frac{D}{2} \log 2\pi, \mu_i^{(l)} \) refers to the \( l \)th dimension of \( \mu_i \), and similarly for \( \sigma_i^{(l)} \).

Note that this symmetric measure can be viewed as the expectation of likelihood of one input’s latent code conditioned on the other, that is

\[ s(x_i, x_j) = \log \int p(z|x_i)p(z|x_j)dz \]

\[ = \log \mathbb{E}_{z \sim p(z|x_j)}[p(z|x_j)] \]  

(4)

\[ = \log \mathbb{E}_{z \sim p(z|x_i)}[p(z|x_i)]. \]

As such, we call it **mutual likelihood score (MLS)**. Different from KL-divergence, this score is unbounded and cannot be seen as a distance metric. It can be shown that the squared Euclidean distance is equivalent to a special case of MLS when all the uncertainties are assumed to be the same:

**Property 1.** If \( \sigma_i^{(l)} \) is a fixed number for all data \( x_i \) and dimensions \( l \), MLS is equivalent to a scaled and shifted negative squared Euclidean distance.

Further, when the uncertainties are allowed to be different, we note that MLS has some interesting properties that make it different from a distance metric:

1. **Attention** mechanism: the first term in the bracket in Equation (3) can be seen as a weighted distance which assigns larger weights to less uncertain dimensions.
2. **Penalty** mechanism: the second term in the bracket in Equation (3) can be seen as a penalty term which penalizes dimensions that have high uncertainties.
3. If either input \( x_i \) or \( x_j \) has large uncertainties, MLS will be low (because of penalty) irrespective of the distance between their mean.
4. Only if both inputs have small uncertainties and their means are close to each other, MLS could be very high.

The last two properties imply that PFE could solve the feature ambiguity dilemma if the network can effectively estimate \( \sigma_i \).

### 4.2. Fusion with PFEs

In many cases we have a template (set) of face images, for which we need to build a compact representation for matching. With PFEs, a conjugate formula can be derived for representation fusion (Figure 5). Let \( \{x_1, x_2, \ldots, x_n\} \) be a series of observations (face images) from the same identity and \( p(z|x_1, x_2, \ldots, x_n) \) be the posterior distribution after the \( n \)th observation. Then, assuming all the observations are conditionally independent (given the latent code \( z \)). It can be shown that:

\[ p(z|x_1, x_2, \ldots, x_{n+1}) = \alpha \frac{p(z|x_{n+1})}{p(z)} \frac{p(z|x_1, x_2, \ldots, x_n)}{p(z|x_1, x_2, \ldots, x_{n+1})}, \]

(5)
where $\alpha$ is a normalization factor. To simplify the notations, let us only consider a one-dimensional case below; the solution can be easily extended to the multivariate case.

If $p(z)$ is assumed to be a noninformative prior, i.e., $p(z)$ is a Gaussian distribution whose variance approaches $\infty$, the posterior distribution in Equation (5) is a new Gaussian distribution with lower uncertainty (See supplementary material). Further, given a set of face images $\{x_1, x_2, \ldots, x_n\}$, the parameters of the fused representation can be directly given by:

$$\hat{\mu}_n = \frac{1}{\sum_{i=1}^n \sigma_i^2} \sum_{i=1}^n \frac{\sigma_i^2 \mu_i}{\sigma_i^2},$$  \hspace{1cm} (6)$$

$$\frac{1}{\sigma_n^2} = \sum_{i=1}^n \frac{1}{\sigma_i^2}. \hspace{1cm} (7)$$

In practice, because the conditional independence assumption is usually not true, e.g., video frames include a large amount of redundancy, Equation (7) will be biased by the number of images in the set. Therefore, we take dimension-wise minimum to obtain the new uncertainty.

**Relationship to Quality-aware Pooling** If we consider a case where all the dimensions share the same uncertainty $\sigma_i$ for $i$th input and let the quality value $q_i = \frac{1}{\sigma_i^2}$ be the output of the network. Then Equation (6) can be written as

$$\hat{\mu}_n = \frac{1}{\sum_{i=1}^n q_i} \sum_{i=1}^n q_i \mu_i \hspace{1cm} (8)$$

If we do not use the uncertainty after fusion, the algorithm will be the same as recent quality-aware aggregation methods for set-to-set face recognition [41, 21, 39].

4.3. Learning

Note that any deterministic embedding $f$, if properly optimized, can indeed satisfy the properties of PFEs: (1) the embedding space is a disentangled identity-salient latent space and (2) $f(x)$ represents the most likely features of the given input in the latent space. As such, in this work we consider a stage-wise training strategy: given a pre-trained embedding model $f$, we fix its parameters, take $\mu(x) = f(x)$, and optimize an additional uncertainty module to estimate $\sigma(x)$. When the uncertainty module is trained on the same dataset of the embedding model, this stage-wise training strategy allows us to have a more fair comparison between PFE and the original embedding $f(x)$ than an end-to-end learning strategy.

The uncertainty module is a network with two fully-connected layers which shares the same input as of the bottleneck layer\(^6\). The optimization criteria is to maximize the mutual likelihood score of all genuine pairs $(x_i, x_j)$. Formally, the loss function to minimize is

$$L = \frac{1}{|P|} \sum_{(i,j) \in P} -s(x_i, x_j) \hspace{1cm} (9)$$

where $P$ is the set of all genuine pairs and $s$ is defined in Equation (3). In practice, the loss function is optimized within each mini-batch. Intuitively, this loss function can be understood as an alternative to maximizing $p(z|x)$: if the latent distributions of all possible genuine pairs have a large overlap, the latent target $z$ should have a large likelihood $p(z|x)$ for any corresponding $x$. Notice that because $\mu(x)$ is fixed, the optimization wouldn’t lead to the collapse of all the $\mu(x)$ to a single point.

5. Experiments

In this section, we first test the proposed PFE method on standard face recognition protocols to compare with deterministic embeddings. Then we conduct qualitative analysis to gain more insight into how PFE behaves. Due to the space limit, we provide the implementation details in the supplementary material.

To comprehensively evaluate the efficacy of PFEs, we conduct the experiments on 7 benchmarks, including the well known LFW [10], YTF [37], MegaFace [13] and four other more unconstrained benchmarks: CFP [28] contains 7,000 frontal/profile face photos of 500 subjects. We only test on the frontal-profile (FP) protocol, which includes 7,000 pairs of frontal-profile faces. IJB-A [18] is a template-based benchmark, containing 25,813 faces images of 500 subjects. Each template includes a set of still photos or video frames. Compared with previous benchmarks, the faces in IJB-A have larger variations and present a more unconstrained scenario. IJB-C [23] is an extension of IJB-A with 140,740 faces images of 3,531 subjects. The verification protocol of IJB-C includes more impostor pairs so we can compute True Accept Rates (TAR) at lower False Accept Rates (FAR). IJB-S [12] is a surveillance video benchmark containing 350 surveillance videos spanning 30 hours in total, 5,656 enrollment images, and 202 enrollment videos of 202 subjects. Many faces in this dataset are of extreme pose or low-quality, making it one of the most challenging face recognition benchmarks (See Figure 2 for example images).

We use the CASIA-WebFace [42] and a cleaned version\(^7\) of MS-Celeb-1M [8] as training data, from which we remove the subjects that are also included in the test datasets\(^8\).

---

\(^6\)Bottleneck layer refers to the layer which outputs the original face embedding.

\(^7\)https://github.com/inlmouse/MS-Celeb-1M_WashList.

\(^8\)4 and 4,182 subjects were removed from CASIA-WebFace and MS-Celeb-1M, respectively.
we show the results on three relatively easier tasks. To be aligned with previous work, we used a different base model, which is a 64-layer

network trained with AM-Softmax on the MS-Celeb-1M dataset. Then we fix the parameters and train the uncertainty module on the same dataset for 12,000 steps. In the following experiments, we compare 3 methods:

- **Baseline** only uses the original features of the 64-layer deterministic embedding along with cosine similarity for matching. Average pooling is used in case of template/video benchmarks.

- **PFE** uses the uncertainty estimation σ in PFE and Equation (6) to aggregate the features of templates but uses cosine similarity for matching. If the uncertainty module could estimate the feature uncertainty effectively, fusion with σ should be able to outperform average pooling by assigning larger weights to confident features.

- **PFEfuse** uses σ both for fusion and matching (with mutual likelihood scores). Templates/videos are fused based on Equation (6) and Equation (7).

In Table 2 we show the results on three relatively easier benchmarks: LFW, YTF and MegaFace. Although the accuracy on LFW and YTF are nearly saturated, the proposed PFE still improves the performance of the original representation. Note that MegaFace is a biased dataset: because all the probes are high-quality images from FaceScrub, the positive pairs in MegaFace are both high-quality images while the negative pairs only contain at most one low-quality image. Therefore, neither of the two types of error caused


to a hyper-spherical embedding space. Then we train the deterministic embeddings, we want to evaluate how it works under different loss functions. In particular, we implement the following state-of-the-art loss functions: Softmax, Triplet Loss, Center Loss, A-Softmax and AM-Softmax.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Data</th>
<th>LFW</th>
<th>YTF</th>
<th>MF1 Rank1</th>
<th>MF1 Veri.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepFace+ [31]</td>
<td>4M</td>
<td>97.35</td>
<td>91.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FaceNet [27]</td>
<td>200M</td>
<td>99.63</td>
<td>95.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DeepID2+ [30]</td>
<td>300K</td>
<td>99.47</td>
<td>93.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CenterFace [35]</td>
<td>0.7M</td>
<td>99.28</td>
<td>94.9</td>
<td>65.23</td>
<td>76.52</td>
</tr>
<tr>
<td>SphereFace [20]</td>
<td>0.5M</td>
<td>99.42</td>
<td>95.0</td>
<td>75.77</td>
<td>89.14</td>
</tr>
<tr>
<td>ArcFace [4]</td>
<td>5.8M</td>
<td>99.83</td>
<td>98.02</td>
<td>81.03</td>
<td>96.98</td>
</tr>
<tr>
<td>CosFace [34]</td>
<td>5M</td>
<td>99.73</td>
<td>97.6</td>
<td>77.11</td>
<td>89.88</td>
</tr>
<tr>
<td>L2Face [25]</td>
<td>3.7M</td>
<td>99.78</td>
<td>96.08</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Baseline</td>
<td>-</td>
<td>99.70</td>
<td>97.18</td>
<td>79.43</td>
<td>92.93</td>
</tr>
<tr>
<td>PFEfuse</td>
<td>4.4M</td>
<td>97.32</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PFEfuse+match</td>
<td>4.4M</td>
<td>97.36</td>
<td>87.95</td>
<td>92.51</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Results of our models (last three rows) trained on MS-Celeb-1M and state-of-the-art methods on CFP (frontal-profile protocol) and IJB-A.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Data</th>
<th>IJB-C (TAR@FAR)</th>
<th>CFP-FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>0.001%</td>
<td>0.01%</td>
<td>0.1%</td>
</tr>
<tr>
<td>Yin et al. [43]</td>
<td>0.5M</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Cao et al. [2]</td>
<td>3.3M</td>
<td>74.7</td>
<td>84.0</td>
</tr>
<tr>
<td>Multicolumn [39]</td>
<td>3.3M</td>
<td>77.1</td>
<td>86.2</td>
</tr>
<tr>
<td>DCN [38]</td>
<td>3.3M</td>
<td>-</td>
<td>88.5</td>
</tr>
<tr>
<td>Baseline</td>
<td>4.4M</td>
<td>70.10</td>
<td>85.37</td>
</tr>
<tr>
<td>PFEfuse</td>
<td>4.4M</td>
<td>83.14</td>
<td>92.38</td>
</tr>
<tr>
<td>PFEfuse+match</td>
<td>4.4M</td>
<td>89.64</td>
<td>93.25</td>
</tr>
</tbody>
</table>

Table 4: Results of our models (last three rows) trained on MS-Celeb-1M and state-of-the-art methods on IJB-C.
by the feature ambiguity dilemma (Section 3) will show up in MegaFace and it naturally favors deterministic embeddings. However, the PFE still maintains the performance in this case. We also note that such a bias, namely the target gallery images being of higher quality than the rest of gallery, would not exist in real world applications.

In Table 3 and Table 4 we show the results on three more challenging datasets: CFP, IJB-A and IJB-C. The images in these datasets present larger variations in pose, occlusion, etc, and facial features could be more ambiguous. As such, we can see that PFE achieves a significant improvement on these three benchmarks. In particular on IJB-C at FAR = 0.001%, PFE reduces the error rate by 64%. In addition, simply fusing the original features with the learned uncertainty (PFE\textsubscript{fuse}) also helps the performance.

In Table 5 we report the results on three protocols of the latest benchmark, IJB-S. Again, PFE is able to improve the performance in most cases. Notice that the gallery templates in the “Surveillance-to-still” and “Surveillance-to-booking” all include high-quality frontal mugshots, which

by the feature ambiguity dilemma (Section 3) will show up in MegaFace and it naturally favors deterministic embeddings. However, the PFE still maintains the performance in this case. We also note that such a bias, namely the target gallery images being of higher quality than the rest of gallery, would not exist in real world applications.

In Table 3 and Table 4 we show the results on three more challenging datasets: CFP, IJB-A and IJB-C. The images in these datasets present larger variations in pose, occlusion, etc, and facial features could be more ambiguous. As such, we can see that PFE achieves a significant improvement on these three benchmarks. In particular on IJB-C at FAR = 0.001%, PFE reduces the error rate by 64%. In addition, simply fusing the original features with the learned uncertainty (PFE\textsubscript{fuse}) also helps the performance.

In Table 5 we report the results on three protocols of the latest benchmark, IJB-S. Again, PFE is able to improve the performance in most cases. Notice that the gallery templates in the “Surveillance-to-still” and “Surveillance-to-booking” all include high-quality frontal mugshots, which

present little feature ambiguity. Therefore, we only see a slight performance gap in these two protocols. But in the most challenging “surveillance-to-surveillance” protocol, larger improvement can be achieved by using uncertainty for matching. Besides, PFE\textsubscript{fuse+match} improves the performance significantly on all the open-set protocols, which indicates that MLS has more impact on the absolute pairwise score than the relative ranking.

5.3. Qualitative Analysis

Why and when does PFE improve performance? We first repeat the same experiments in Section 3 using the PFE representation and MLS. The same network is used as the base model here. As one can see in Figure 6, although the scores of low-quality impostor pairs are still increasing, they converge to a point that is lower than the majority of genuine scores. Similarly, the scores of cross-quality genuine pairs converge to a point that is higher than the majority of impostor scores. This means the two types of errors discussed in Section 3 could be solved by PFE. This is further confirmed by the IJB-A results in Figure 7. Figure 8 shows
the distribution of estimated uncertainty on LFW, IJB-A and IJB-S. As one can see, the “variance” of uncertainty increases in the following order: LFW < IJB-A < IJB-S. Comparing with the performance in Section 5.2, we can see that PFE tends to achieve larger performance improvement on datasets with more diverse image quality.

**What does DNN see and not see?** To answer this question, we train a decoder network on the original embedding, then apply it to PFE by sampling \( z \) from the estimated distribution \( p(z|x) \) of given \( x \). For a high-quality image (Figure 9 Row 1), the reconstructed images tend to be very consistent without much variation, implying the model is very certain about the facial features in this images. In contrast, for a lower-quality input (Figure 9 Row 2), larger variation can be observed from the reconstructed images. In particular, attributes that can be clearly discerned from the image (e.g. thick eye-brow) are still consistent while attributes cannot (e.g. eye shape) be discerned have larger variation. As for a mis-detected image (Figure 9 Row 3), significant variation can be observed in the reconstructed images: the model does not see any salient feature in the given image.

### 6. Risk-controlled Face Recognition

In many scenarios, we may expect a higher performance than our system is able to achieve or we may want to make sure the system’s performance can be controlled when facing complex application scenarios. Therefore, we would expect the model to reject input images if it is not confident. A common solution for this is to filter the images with a quality assessment tool. We show that PFE provides a natural solution for this task. We take all the images from LFW and IJB-A datasets for image-level face verification (We do not follow the original protocols here). The system is allowed to “filter out” a proportion of all images to maintain a better performance. We then report the TAR@FAR = 0.001\% against the “Filter Out Rate”. We consider two criteria for filtering: (1) the detection score of MTCNN \[ 36 \] and (2) a confidence value predicted by our uncertainty module. Here the confidence for \( i \)th sample is defined as the inverse of harmonic mean of \( \sigma_i \) across all dimensions. For fairness, both methods use the original deterministic embedding representations and cosine similarity for matching. To avoid saturated results, we use the model trained on CASIA-WebFace with AM-Softmax. The results are shown in Figure 11. As one can see, the predicted confidence value is a better indicator of the potential recognition accuracy of the input image. This is an expected result since PFE is trained under supervision for the particular model while an external quality estimator is unaware of the kind of features used for matching by the model. Example images with high/low confidence/quality scores are shown in Figure 10.

### 7. Conclusion

We have proposed probabilistic face embeddings (PFEs), which represent face images as distributions in the latent space. Probabilistic solutions were derived to compare and aggregate the PFE of face images. Unlike deterministic embeddings, PFEs do not suffer from the feature ambiguity dilemma for unconstrained face recognition. Quantitative and qualitative analysis on different settings showed that PFEs can effectively improve the face recognition performance by converting deterministic embeddings to PFEs. We have also shown that the uncertainty in PFEs is a good indicator for the “discriminative”quality of face images. In the future work we will explore how to learn PFEs in an end-to-end manner and how to address the data dependency within face templates.
References


